REMARKS

Claims 1-4 and 6-10 have been amended. New claims 11 and 12 have been added. Accordingly, claims 1-12 are currently pending.

Priority

Applicants appreciate the Examiner's acknowledgment of the claim for priority. Submitted herewith is the certified copy of the priority document, JP 2000-152665. An indication that the priority document has been received would be appreciated.

35 U.S.C. §102

Claims 1-10 have been rejected under 35 U.S.C. § 102(b) as being anticipated by Lauterbach, "Trace32 RTOS Debugger for pSOS+", hereinafter TRACE32. Applicants request reconsideration of the rejection in view of the amendments to the claims and for the following reasons.

The present invention is directed to a multi-process display method of displaying processes to be debugged in a debugger system and to a debugger system as well as a program

storage device storing a program of instructions executable to perform steps for debugging multi-processes. Conventionally, a debugger provides two functions. One is a stop-check function wherein the debugger can stop the program execution at a break point designated by the user, such as a certain address or a certain event. Then, while the program is stopped, the user can check the contents of the registers and the memories at the break point on the display device through the debugger. The other typical function provided by a debugger is a statistical analysis function. Information on resources used by each of the executed programs and the system behavior is collected by the debugger during the execution of the programs. Such information may be the CPU execution time, the consumed stack space, etc. After execution of the program, the user can refer to the collected information and the results of the statistical analysis.

The TRACE32 system shows a debugger that provides the stop-check and statistical analysis functions. The stop-check function as shown in the figures of pages 2 and 3, and the statistical analysis function is shown in the figures of pages 4, 5 and 6 of the reference. Referring to the figure on page

2 of the reference, the operation state of each task and the register contents at the break point (see the first line in the figure) are shown. However, the TRACE32 system does not disclose updating the display to indicate the operation state of each process when detecting the operation state change, as claimed by Applicants in claims 1, 8 and 10, which are the independent claims.

According to the present invention, the program operation state is displayed, for example, and the display is updated to indicate the operation state of each process when detecting the operation state change. In particular, referring to Fig. 1 and pages 5 and 6 of the specification, when an operation state of the debug process 101 changes, the debug execution processing section 121 transfers control to the process operation information output section 122. A process operation information monitoring section 141 checks to determine the presence or absence of the output of the process operation information 131, which is received from the output section 122. A process information display section 144 refers to a process information table created by section 143 and displays

the process graph 151 and process detail information 152 on the display 150.

Each process constituting the program of the present invention can be represented in a process graph 151 by a process box or rectangle, for example, as shown in Fig. 8. The relationship between the processes, such as the parent and child, and brothers is represented by the topology of the rectangles, displayed in a family tree. The operation state is displayed graphically (151) or in tabular form (152). Further, the display is automatically updated every time the debugger detects a change in the operation state of a process.

In the TRACE32 system, the task list is displayed with the operations status, as shown on pages 2 and 3 of the reference. The task list does not indicate the relationship between the tasks. The figure on page 5 labeled "statistics and flow of tasks" shows how the CPU executes plural tasks while switching them. As described at the top of page 5, this figure represents statistical evaluation and graphical display of task run times. The system obtains data of task switching and the CPU execution time of each task and displays this information on a time axis basis on the data that has been

obtained. Accordingly, the reference merely discloses one of the functions generally provided by a debugger, e.g., the statistical analysis function. The graphic display shown by the figures of the reference does not show the current operation status indicating the operation state of each process when detecting the operation state change as set forth in claim 1, for example. Accordingly, the reference does not anticipate the invention as claimed and the 35 U.S.C. §102 rejection should be withdrawn.

Conclusion

In view of the foregoing amendments and remarks, Applicants contend that the above-identified application is now in condition for allowance. Accordingly, reconsideration and reexamination is requested.

Respectfully submitted,

John R. Mattingly
Registration No. 30,293
Attorney for Applicant(s)

MATTINGLY, STANGER & MALUR
1800 Diagonal Rd., Suite 370
Alexandria, Virginia 22314
(703) 684-1120
Date: September 21, 2004

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Commissioner For Patents and Trademarks, Alexandria, VA 22313-1450

on Sept 21, 2004, by